13. ročník
Žilina 1. – 3. 7. 2025
Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach
ossconf.fri.uniza.sk
OSS Conf

OSSConf 2025: 1–8

Optimalizujeme s programem Concorde

PAVEL STŘÍŽ (CZ)

Abstrakt. Článek představuje základní užití programu Concorde. Jako samostatně spustitelný soubor, v programu R přes knihovnu `tsp` a přes NEOS server (v R a Pythonu). Na kresbu je užit příkaz `plot` a `ggplot`. NEOS server užívá GnuPlot. Velmi stručně jsou zmíněny alternativy TSP a možné práce do budoucna.

Klíčová slova. Concorde, optimalizace, R, Linux.

OPTIMIZATION WITH THE CONCORDE TSP SOLVER

Abstract. The article introduces basic use of the Concorde TSP Solver, as a standalone executable, in R with the help of the `tsp` library and via the NEOS server (R and Python interface). Drawing is done with `plot` and `ggplot` commands. The NEOS server is using GnuPlot. It also briefly introduces some TSP alternatives and future plans.

Keywords. Concorde, optimization, R, Linux.

1. Pár přejivých slov místo Úvodu

V roce 2009 (16 let nazpět v době psaní článku) mě tehdejší kolega Libor Sarga poprosil, jestli bych mu nepomohl se získáním vektorových obrázků u programu Concorde, `https://www.math.uwaterloo.ca/tsp/concorde`, pro připravovanou kapitolu v knize. Je to program řešící NP-úplně těžký problém obchodního cestujícího (Traveling Salesman Problem, TSP). Učil jsem operační výzkum, tak mi to přišlo jako zajímavý sazební úkol. Tehdy jsem nepracoval s Linuxem přímo, jen se CygWinem, tak jsem si na práci z příkazového řádku či hlubší pokusy netroufl. Objevil jsem však možnost uložit si výsledky do textových souborů v grafické verzi programu pod Microsoft Windows. Vznikla primitivní verze v Bashi.

Tehdy už byla Lua v TEXovém světě docela zaběhlá, tak jsem zkusil tu. Vznikla tato Q+A na TEX.SE, `https://tex.stackexchange.com/questions/235372/how-to-get-vector-graphics-from-concorde-program`. To se psal rok 2015 (10 let nazpět).

Shodou okolností se jeden z autorů, profesor Vašek Chvátal, vrátil z Kanady do Prahy a udělal jsem s ním rozhovor, viz tento sborník. Profesor se omlouvá, že na OSSConf nedorazí, ale ví o nás. Napadlo mě ty mé TEXové pokusy kolem Concorde oprášit, možná až do podoby TEXového balíčku. Tak jsem to začal zkoumat s novými programátorskými zkušenostmi.

2. Instalace

Vynechám instalační část pro Microsoft Windows, to jsem si kdysi zkusil a fungovalo to. Vrhnu se přímo do pro mne něčeho nového. Nahození v Linuxu bylo překvapivě rychlé. Ve složce jsem provedl tyto kroky. Plus lze nastavit složku do PATH, uživatelé Linuxu ví.

```
wget -c https://www.math.uwaterloo.ca/tsp/concorde/downloads/
   codes/linux24/concorde.gz
gzip -d concorde.gz
chmod +x concorde
./concorde
```

Podobně lze nahodit program Linkern, který za optimalizaci jde na věci jinak.

```
wget -c https://www.math.uwaterloo.ca/tsp/concorde/downloads/
   codes/linux24/linkern.gz
gzip -d linkern.gz
chmod +x linkern
./linkern
```

Vše se zdálo funkční.

Vyzkoušel jsem si i instalaci ze zdrojových kódů, to není nikdy na škodu, protože zásahy do zdrojových kódů člověk potřebuje častěji než by si to připouštěl.

V nové složce jsem začal:

```
wget -c https://www.math.uwaterloo.ca/tsp/concorde/downloads/
   codes/src/co031219.tgz
tar xvf co031219.tgz
cd concorde
```

Plus jsem stáhl soubory pro QSopt, jak radí instalační manuál.

```
mkdir -p QS; cd QS
wget -c https://www.math.uwaterloo.ca/~bico/qsopt/downloads/
   codes/ubuntu/qsopt{,.a,.h}
cd ..
```

Následovaly poslední dva příkazy:

```
./configure --with-qsopt=${PWD}/QS --with-cplex=${PWD}/LP
make
```

3. Základní užití

A hned jsem se pustil do testů. Manuál radí si zkusit 100 bodů s nastavením random seed (počáteční hodnota generátoru pseudonáhodných čísel).

```
cd TSP
./concorde -s 99 -k 100
./concorde -s 99 -k 100 -o test100a.sol
```

```
./concorde -s 99 -k 100 -f -o test100b.sol
```

První řádek vygeneruje 100 bodů a najde okružní cestu. Druhý řádek uloží výsledky do souboru, ale jen trasu. Poslední řádek to samé, ale trasa je vypsána nejen jako uzly, ale i s hranami. Soubory jsou textové, dá se s nimi dále pracovat. To bylo rychlé, že?

4. TSPLIB95

Abychom nepracovali se simulovanými daty, existuje krásná webovka s testovacími daty.

```
cd ..; mkdir -p TSPLIB95; cd TSPLIB95
wget -c http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/
   tsp/ALL_tsp.tar.gz
tar xvf ALL_tsp.tar.gz
for soubor in ‘find -iname \*.gz‘; do echo $soubor; gzip -d $soubor; done
cd ../TSP
```

Nu, a můžeme si rovnou vyzkoušet vyřešit jeden z problémů.

```
./concorde ../TSPLIB95/a280.tsp
```

Výhoda těchto dat je, že lze nahlédnout na optimální cestu. To jsou soubory s příponou `.opt.tour`. A jsme doma!

Přes Python. Uživatelům vřele doporučuji `https://tsplib95.readthedocs.io`. Nezávislé testy jsou nad rámec tohoto příspěvku.

5. R

Zkoumal jsem, jestli si lze Concorde spustit v R.

Pokud nemáme, nainstalujeme si R. Podrobné návody lze nalézt na `https://cran.r-project.org/`. Spustíme si R a doinstalujeme knihovnu `TSP`. Detaily viz `https://cran.r-project.org/web/packages/TSP/vignettes/TSP.pdf`.

```
$ R
# install.packages("TSP")
```

Knihovnu si načteme, nastavíme cestu ke spustitelnému souboru a můžeme zkusit funkčnost knihovny. Asi nejrychlejší cesta, jak si zjistit cestu k adresáři je `echo ${PWD}`.

```
library(TSP)
concorde_path("...") # Nutné si upravit.
data("USCA312")
solve_TSP(USCA312, method="concorde", control=list(clo="-v"))
```

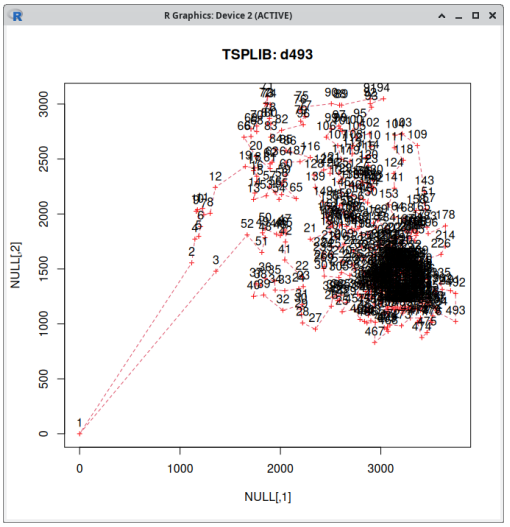
Knihovna `TSP` jednu ukázku z TSPLIB95 obsahuje, můžeme rovnou zkoušet. Obrázek vlevo na další straně.

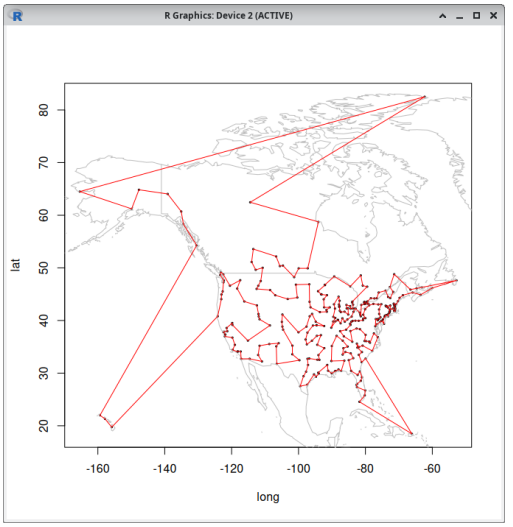
```
(drill<-read_TSPLIB(system.file("examples/d493.tsp",package="TSP")))
```

```
(tour<-solve_TSP(drill,method="concorde")) # Bylo nn,two_opt=TRUE.
plot(drill, tour, cex=.6, col = "red", pch= 3, main = "TSPLIB: d493")
```

Zkusíme si ještě jednu ukázku z manuálu jako vzorek práce se skutečnou mapou. Obrázek vpravo.

```
# install.packages("maps")
library(maps)
data("USCA312")
(tour<-solve_TSP(USCA312, method="concorde"))
data("USCA312_GPS")
head(USCA312_GPS)
plot((USCA312_GPS[, c("long", "lat")]), cex = .3)
map("world", col = "gray", add = TRUE)
polygon(USCA312_GPS[, c("long", "lat")][tour,], border = "red")
```





Zájemcům o mapy přes R lze na další bádání minimálně doporučit `https://ggplot2-book.org/maps.html`. Více o tom zvaná přednáška Michala Páleníka.

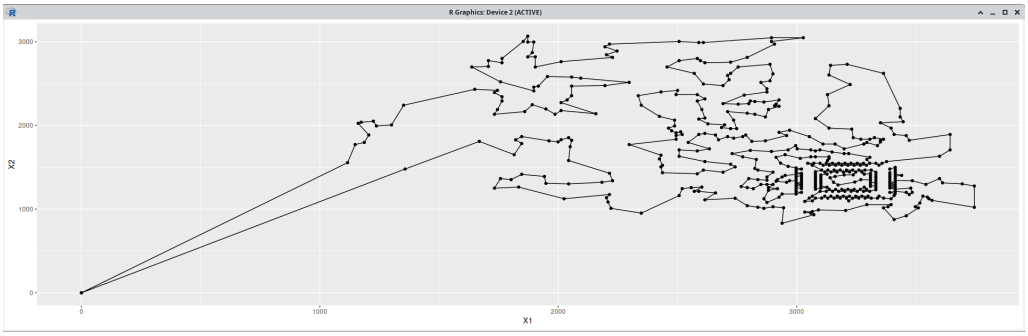
6. R: ggplot2

Po bližším zkoumání jsem zjistil, že TSP sice používá Concorde, ale nějakým způsobem zasahuje do .tsp souboru před vyřešením, byť mi funkce `read_TSPLIB` a `write_TSPLIB` generovaly identický soubor. Jedná se konkrétně o to, že se snaží eliminovat desetinná místa, ale to jen před funkcí `solve_TSPLIB`.

Docházelo ke změně z `3 1.35760e+03 1.47900e+03` na `3 1e+03 1e+03`. Nedokázal jsem to obejít. Zahlásil jsem to autorovi knihovny (verze 1.2-4). Opravil to 27. května 2025, `https://github.com/mhahsler/TSP/issues/21`, vznikla verze 1.2-5. Svět jsem sice nespasil, ale každý zlepšovák potěší!

Zde je naznačení cesty, jak výsledky vykreslit přes knihovnu `ggplot2`.

```
# install.packages(c("dplyr","data.table","ggplot2"))
library(dplyr, data.table, ggplot2)
maldrill<-data.frame(drill); head(maldrill)
maltour<-data.frame(tour); head(maltour)
maldrill$poradi<-cbind(c(maltour$tour))
A<-maldrill[c(maldrill$poradi),]
B<-as.data.table(A)
C<-rbind(B,B[1])
ggplot(data=C,aes(x=X1,y=X2))+geom_point()+geom_path()
```



Poznámka 1. Finta s tabulátorem funguje i v R. Začneme psát příkaz a dáme Tab, jsou vidět dostupné možnosti.

Poznámka 2. Člověk to bral jako samozřejmost, předponu „gg“. Je to odkaz na knihu *The Grammar of Graphics*, `https://link.springer.com/book/10.1007/0-387-28695-0`.

Poznámka 3. Ony plusy mezi funkcemi není sčítání, ale vrstvení grafu. Jinými slovy přidávání podkladů do grafu.

7. Server NEOS

Poměrně nedávnou záležitostí je možnost využít komerční i nekomerční řešitele přes tento server. Dává to možnost si řadu věcí vyzkoušet, není třeba si programy instalovat či kupovat. Omezení jsou zřejmá: výpočet musí skončit do určité doby (je to závislé od programu, počítejte v řádu hodin), musíte zadání v nějakém formátu na server dostat, pak si stáhnout zpět atd. Pro výuku ideální záležitost.

Mezi řešiteli se objevil i program Concorde.

Mé první pokusy se starší knihovnou `rneos` nepřinesly žádané výsledky. Problém byl, že email je nově povinný parametr a je potřeba seznam parametrů upravit vůči ukázce z dokumentace.

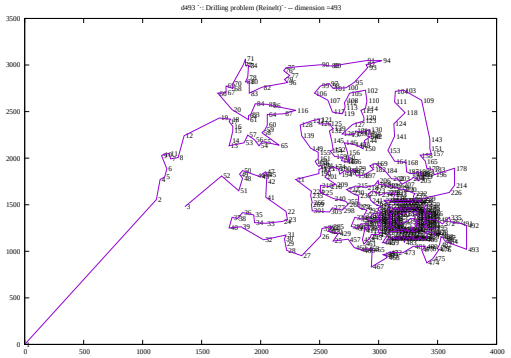
```
# install.packages("rneos")
library(rneos)
NlistSolversInCategory(category = "co")
template <- NgetSolverTemplate(category = "co", solvername = "concorde",
   inputMethod = "TSP")
```

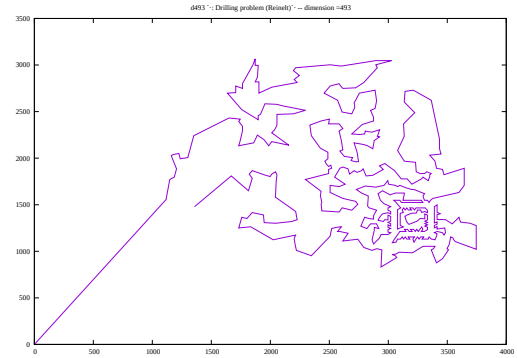
```
modc <- paste(paste(readLines("d493.tsp"), collapse = "\n"), "\n")
argslist <- list(tsp=modc, comment="", email="váš-email", ALGTYPE="con",
   RDTYPE="fixed", PLTYPE="cp")
xmls <- CreateXmlString(neosxml = template, cdatalist = argslist)
NprintQueue()
(test <- NsubmitJob(xmlstring=xmls, user="váš-email", interface=""))
NgetJobStatus(obj = test)
NgetFinalResults(obj = test)
```

Zkusil jsem i novější knihovnu `roi`, tam se pracuje s pluginy. Plugin pro Concorde nevidím, napsal jsem tip autorovi. Seznam pluginů a jejich instalace je rozepsána na `http://roi.r-forge.r-project.org/installation.html`.

Pro náš test stačí webové rozhraní. Navštívíme `https://neos-server.org`, dáme Submit a job to NEOS, dohledáme concorde a naklikneme TSP. Dále vybíráme Browse... pod Concorde data (TSPLIB format file), a užijeme ukázku z R: `~/R/x86_64-pc-linux-gnu-library/4.4/TSP/examples/d493.tsp`. Můžeme si nastavit další parametry a volíme Submit to NEOS. Za pár okamžiků dorazí výsledek, buď na webové stránce či přes email. Já jsem na serveru registrovaný, takže mohu zvolit pod svým jménem My Jobs a vidím seznam úloh. Volím příslušný řádek ve sloupci Results.

U prvních testů byla na posledním řádku informace, že lze stáhnout PDF, resp. ZIP verzi, mapy. Ale nebylo to klikatelné. Hans Mittelmann, tvůrce rozhraní Concorde na NEOSu, psal, že administrátoři NEOS serveru zrušili možnost sdílet binární soubory, kam spadá i PDF. Zvažoval tu možnost zrušit. Na mou přímluvu zařadil PostScriptový soubor. Podařilo se tak 27. května 2025. Zde jsou ukázky, jak na to jde. V pozadí spouští GnuPlot.





Kdo se nerad pročítá XML specifikacemi a má rád hackování, tak jeden tip. Lze si ve spodní části navolit Dry run. To vygeneruje XML, uložíme si, upravíme `<email>` a soubor `zaslat.xml` zkusíme podstrčit Pythonu, šly by určitě i jiné jazyky.

```
pip3 install neos_client --break-system-packages
```

U seriózní práce opět užijeme virtuální prostředí. A vlastní test by vypadal:

```
from neos_client import NeosClient
client=NeosClient(email="váš-email-na-NEOS-serveru")
client.submit_job(xml_path="zaslat.xml")
```

Za pár okamžiku dorazí číslo úlohy a heslo a u naší ukázky za pár vteřin se vypíší výsledky a dorazí email. Výstupy jsou ve třech sloupcích. Ze kterého uzlu a do kterého uzlu se spojujeme a na celá čísla zaokrouhlená vzdálenost.

Lze si z takového souboru rychle udělat šablonu, to už nechávám na laskavém čtenáři k bádání. V principu si vzít první řádky, vpasovat vlastní data a napojit poslední řádky. Struktura je:

```
<document>
<category>co</category>
<solver>concorde</solver>
<inputMethod>TSP</inputMethod>
<email> insert valid email address </email>
<dat2><![CDATA[...Insert Value Here...]]></dat2>
<dat1><![CDATA[...Insert Value Here...]]></dat1>
<tsp><![CDATA[...Insert Value Here...]]></tsp>
<ALGTYPE><![CDATA[...Insert Value Here...]]></ALGTYPE>
<RDTYPE><![CDATA[...Insert Value Here...]]></RDTYPE>
<PLTYPE><![CDATA[...Insert Value Here...]]></PLTYPE>
<comment><![CDATA[...Insert Value Here...]]></comment>
</document>
```

Volí se mezi běžnou vzdáleností `dat2`, Manhattan `dat1` či formát TSPLIB. V ALGTYPE se volí `con` (CPLEX), `cqs` (QSort) či `lk` (Lin-Kernighan). V RDTYPE, počáteční hodnota generátoru pseudonáhodných čísel, se volí `fixed` (99) či `variable` (náhodně). V PLTYPE pro vykreslení: `no` (žádné), `cp` (nalezená trasa s uzly) nebo `pf` (bez uzlů).

Výzva. Jak úlohu distribuovat na víc výpočetních strojů. To jsem zatím vzdal.

8. Varianty TSP

Existuje několik zajímavých variant problému obchodního cestujícího. Za prolistování stojí webové stránky TSPLIB95, `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/`. S těmi nemá Concorde problém, např. asymetrická varianta TSP, uzly definované jako body v prostoru ap.

Pamatuji si variantu TSP ještě z dob studií. U TSP je každý uzel statický, představme si však, že každý uzel má svou trajektorii. Je to známá úloha sestřelení vesmírného odpadu, které jsou na oběžné dráze Země. Pracovně: Dynamický TSP.

Během příprav článků do tohoto sborníku se vyskytla ještě jedna varianta. Vezměme si mapu ČR a SR, například kraje. Řešme TSP a chceme navštívit všechny kraje. Jenže kraj nebude zjednodušen jako uzel (těžiště, administrativní centrum či nějaký zvolený bod), ale je to oblast/polynom, kde se stačí dotknout hranice na libovolném místě. Pracovně: Polynomiální TSP.

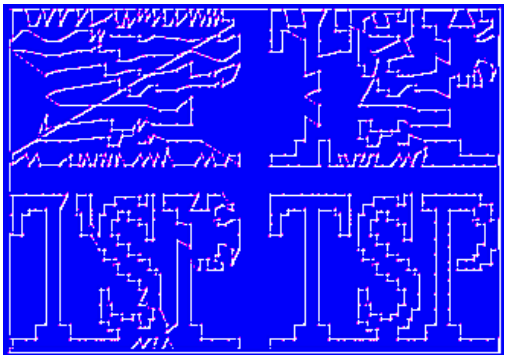
9. Vytyčené úkoly místo Závěru

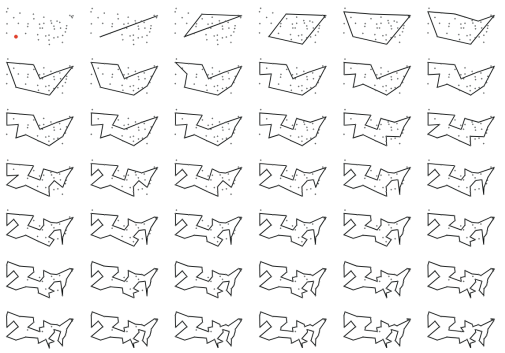
Koho by tato oblast zajímala víc, doporučuji přednášku profesora Chvátala, `https://slideslive.com/38895310`, a jejich knihu *The Travelling Salesman Problem: A Computational Study* z roku 2007. To je hutnější čtení, kdo by rád o něco jemnější, lze doporučit knihu Williama Cooka *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*, podpořenou přednáškou `https://slideslive.com/39000207`.

Mé starší pokusy, jak získat vektorové obrázky z Concorde, už zastaraly. Jak je vidět, program R to má vymazlené. Pro mne jsou tu typografické výzvy:

- Čísla na hranách se sází s bílým kruhem či obdélníkem. To by chtělo obejít tak, že se hrana vysází jako dvě úsečky.
- Aby se čísla na hranách nepřekrývaly, nabízí se možnost posunout číslo ze středu hrany kousek jinam.

Oba vytyčené úkoly by stálo za hřích vyřešit a nechat spočítat, jen to před OSSConf 2025 nestihnu. Kdyby to bylo hezké, dalo by se to uvážit poslat Hans Mittelmannovi jako výstup z NEOS serveru.





Je zde ještě jedna výzva. Pokud se podíváme na logo TSPLIB95 nebo nahlédneme na str. 66–74 Cookovy knihy, je vidět průběh řešení TSP. To by bylo vhodné z Concorde získat též jako studijní pomůcku. To teprve zkoumám možnosti realizace. Štefan Peško upozorňuje, že ne každý výpočetní krok by se odrazil na mapě. Je to na uváženou.

V článku jsme si ukázali vykreslení TSP ve 2D, ale Concorde umí i 3D. To by byla výzva za vykreslení, např. přes program Raylib.

Kontaktní adresa

Ing. Pavel Stříž, Ph.D., Nakladatelství Martin Stříž, U Škol 940, Bučovice, okres Vyškov, 685 01, Česká republika,
E-mailová adresa: `pavel@striz.cz`